AIAA Guidance, Navigation, and Control Conference
13 - 16 August 2012, Minneapolis, Minnesota

AIAA 2012-4975

Towards a Fast-time Simulation Analysis of Benefits of the Spot and Runway Departure Advisor

Robert D. Windhorst[*]
NASA Ames Research Center, Moffett Field, CA, 94035

NASA Ames Research Center is developing a concept for managing flight operations on airport surfaces. The goal of the concept, named the Spot and Runway Departure Advisor (SARDA), is to reduce delays, emissions, noise, and fuel consumption. In 2010, human-in-the-loop simulations of the SARDA concept were conducted.[1] Results showed that for the 2008 heavy traffic scenario SARDA reduced departure delays and fuel consumption, while imposing little impact on perceived controller workload. The 2008 normal traffic scenario did not show measurable benefits. The human-in-the-loop simulations analyzed only two traffic scenarios and were costly. To efficiently expand the research of SARDA so that it included analysis of more traffic scenarios, adaptation to more airports, and investigation of changes to the concept, a fast-time simulation capability was needed. This paper documents the first attempt at fast-time simulation analysis of the SARDA concept. A fast-time simulation of traffic at Dallas/Fort Worth International Airport being autonomously managed by a delay-optimal runway scheduler was conducted. Although the simulation was meant to analyze the benefits of the SARDA concept, there were differences between how traffic was managed in the fast-time simulation and the human-in-the-loop simulations. The differences were due to difficulties adapting the delay-optimal scheduler from the human-in-the-loop simulations to the fast-time simulation and lack of capabilities of the airport operations model. The main differences were that the human-in-the-loop simulations included runway crossings in the optimal departure schedules, while the fast-time simulation did not, and the human-in-the-loop simulations controlled flights at the spot, while the fast-time simulation controlled flights at the gate. Results of the delay-optimal fast-time simulation showed benefits for the 2008 normal traffic schedule. It produced 8% less average taxi time and 40% fewer stops than a simulation controlled by a first-come-first-served scheduler. However, the delay-optimal simulation had the same average delays as the first-come-first-served simulation. In terms of equity of flight delays, the worst delayed flight in the delay-optimal simulation had one and a half minutes more delay than the worst delayed flight in the first-come-first-served simulation.

Nomenclature

variables		
a	=	aircraft acceleration
s	=	aircraft path length
t	=	time
v	=	aircraft speed

acronyms		
GRP	=	Gate Release Planner
RS	=	Runway Scheduler
SARDA	=	Spot and Runway Departure Advisor
SOSS	=	Surface Operations Simulator and Scheduler
SRP	=	Spot Release Planner
TFS	=	Tactical Flight Separation

[*] Aerospace Engineer, Aerospace High Density Operations Branch, NASA Ames Research Center/Mail Stop 210-10, Senior member.

I. Introduction

TRAFFIC on the surface of busy airports in the United States accounts for a large part of national flight delays, noise, and emissions.[2] Generally, flights push back from their gates and taxi to a queue of flights waiting to use the runway. While the flights are queued, they burn fuel, release emissions, and produce noise. Several methods for reducing this wait time are described in the literature. One method is to keep the queue as short as possible without drying out operations on the runway by holding flights at their gates or spots. Another method is to maximize the usage of the runway by optimally sequencing operations on the runway. At Boston Logon International Airport, Simaiakis et.al.[3-4] field tested a concept for metering the number of flights allowed to push back from the gate onto the airport taxiway. Results showed that the concept was able to save 3,900-4,900 US gallons of fuel during four eight-hour tests by decreasing the taxi time, with a increase in gate departure delay of 4.3 minutes per flight on average. Brinton et. al.[5] developed a Collaborative Departure Queue Management concept. This concept uses the ration-by-schedule algorithm that is used in Ground Delay Programs to assign flight-by-flight departure slots. The concept was field tested at Memphis International Airport. Initial results showed that the concept was able to provide slot times that controlled the departure queue length. Although these concepts produced benefits, their departure sequences were not guaranteed to be minimum-delay sequences, so other methods may be able to produce even greater benefits.

An algorithm for generating minimum-delay and maximum-throughput departure sequences has been developed using the basic principles of mixed integer linear programming[6-10] and of dynamic programming.[11] The dynamic programming version produced in less execution time similar results to the mixed integer linear programming version. In a standalone analysis, departure sequences produced by the dynamic programming version had 43% less delay and 1.3% more throughput than those produced by another algorithm that modeled the human controller.[12] This algorithm was integrated into a concept for managing surface traffic named the Spot and Runway Departure Advisor (SARDA), which is being developed at NASA Ames Research Center.[13] In 2010, high-fidelity human-in-the-loop (HITL) simulations of the concept managing traffic on the east side of Dallas/Fort Worth International airport were conducted. Initial results for the heavy scenario studied showed that the concept reduced departure delays by 64% and fuel consumption on the ground by 38%, while imposing little impact on perceived controller workload.[1&14] There were little changes in the performance metrics for the normal scenario that was studied.

The HITL simulations analyzed two traffic scenarios at a single airport. In addition, the simulations analyzed a single version of the SARDA concept. To expand research of the SARDA concept to include more traffic scenarios and airports and to investigate how changes to the concept would affect benefits, NASA developed a fast-time simulation of airport surface traffic named the Surface Operations Simulator and Scheduler (SOSS).[15] The purpose of SOSS is to efficiently test and analyze concepts for managing traffic on airport surfaces.

In this paper, SOSS was used to conduct a fast-time simulation of a delay-optimal scheduler managing traffic on the east side of Dallas/Fort Worth International Airport. This simulation was an initial step toward using SOSS to analysis the benefits of SARDA. Although the delay-optimal scheduler was the same as the one used in the 2010 HITL simulation of SARDA, the way the scheduler was applied to managing traffic was different. In the 2010 HITL simulation, flights crossing the runway were included in the optimal schedule, while in the fast-time simulation crossing flights were not included in the optimal schedule. In addition in the 2010 HITL simulation, flights were cleared from the spot, while in the fast-time simulation, flights were cleared from the gate. A summary of all the differences is provided in the next section. The differences were due to time limitations in adapting the scheduler from the HITL simulation to SOSS and lack of capabilities of SOSS. Performance in terms of delay, number of stops, and taxi time of the delay-optimal simulation was compared with the performance of a simulation of a first-scheduled-first-served scheduler managing traffic. In addition, equity in terms of which simulation had the most delayed flight was studied.

This paper is organized as follows. First, the 2010 HITL SARDA simulation is described and compared with the fast-time simulation. Then, the details of the fast-time simulation tools and algorithms are explained, and the results are presented. The paper ends with conclusions.

II. Spot and Runway Departure Advisor

This section describes the Spot and Runway Departure Advisor (SARDA) concept as it was simulated during the 2010 HITL simulation. First, airport surface locations and procedures that are used extensively in SARDA are explained. Then, SARDA is presented. Finally, a summary of the differences between the SARDA 2010 HITL simulations and the fast-time simulations is provided.

At Dallas/Fort Worth International Airport, spots are locations on the surface where responsibility for control of flights passes from the ramp controller, who works for the airline, to the ground controller, who works for the FAA.

To pass through the spot and enter the taxiway, pilots must obtain clearance from the ground controller. The ground controller maintains control over flights in the taxiway until they enter the departure queue. The departure queue is the area just before the runway entrance where flights line up to takeoff. As flights enter the departure queue, the ground controller hands them off to the local controller. The local controller is responsible for enforcing safe operations on the runway. He clears departing, crossing, or arriving flights to enter and use the runway. Flights wait their turn in the departure queues or at the runway crossings until the local controller gives them clearance, at which time they may use the runway.

The Spot and Runway Departure Advisor (SARDA) is a NASA developed concept for managing surface traffic. Its goal is to reduce taxi times and delay and, thereby, reduce fuel-burn, noise, and emissions. It accomplishes this by providing advisories to ramp, ground, and local controllers. Ground controllers use their advisory to clear flights from the spot, and local controllers use their advisory to sequence flight operations (departures and crossings) on the runway. The focus of SARDA to date as been on decision support tools for controllers. However, the long term vision for SARDA is that it also include a decision support tool for the airline ramp controller, which will be used to manage traffic in the ramp area.

SARDA provides the ground controller with an advisory of when to clear flights at the spots to enter the taxiway. The advisories are designed to meter the number of flights entering the taxiway such that traffic on the taxiway is low and the departure queues are as small as possible without letting the runway operations dry out. For example, subject matter experts estimate the proper number of aircraft in the queue at Dallas/Fort Worth International Airport to be about five flights distributed across three queues that feed runway 17R. This prevents excessive flight taxi times, emissions, and noise.

In addition to metering flights entering the taxiway, the spot clearances are also designed to satisfy a minimum-delay sequence of operations on the runway, where delay is defined as the difference between the flight's scheduled runway use time and the time it would have used the runway if there was no traffic. The sequence includes flight departures, arrivals, and runway crossings and is calculated by the SARDA tools. Although the flights are released from the spot in the delay-optimal sequence to use the runway, unexpected events during taxi (such as conflicts with other flights) can change the sequence. A goal of SARDA research, not addressed in this paper, is to understand how these events degrade the benefits of a delay-optimal schedule.

SARDA provides the local controller with an advisory of the runway operations sequence. The local controller can use this sequence to decide which flight to allow to use the runway next. Depending on the runway, the local controller may have several options. In the south flow airport configuration at Dallas/Fort Worth International Airport for example, runway 17R has three primary departure queues and ten crossing locations. Flights may be waiting in any of these areas to use the runway. The advisory helps the controller identify the next aircraft to use the runway that will minimize delays.

The SARDA concept used in the 2010 human-in-the-loop simulation calculated ground and local controller advisories with two programs: the runway scheduler and the spot release planner. The runway scheduler (RS) generated the minimum-delay sequences of operations on the runway, and the spot release planner (SRP) generated the spot clearance times.[13] Both the RS and SRP had their own implementation of a delay-optimal scheduler. The delay-optimal scheduler calculated, using a dynamic programming approach, minimum-delay departure schedules for the runway. The RS sequence advisory and SRP spot clearance advisory were generated from independent calculations of minimum-delay departure schedules.

The delay-optimal scheduler was adapted from the RS and the SRP to work with SOSS. Due to time constraints to complete the adaptation and limitations of SOSS, the scheduler used in this paper was not configured exactly as the SARDA concept in the 2010 HITL. The scheduler produced flight clearances at the gate, instead of at the spot, which is the case for the 2010 HITL. Since clearances were produced at the gate, the scheduler is hereafter referred to as the gate release planner (GRP).

Another difference between the GRP fast-time simulations and the HITL was that in the HITL the local controller was asked to follow an advisory given by the RS that specified the sequence that flights should use the runway. In the GRP, an optimal sequence was not enforced at the runway entrances. Flights were allowed to use the runway on a first-available-first-served basis.

There were several additional differences between the GRP simulation and the HITL simulations. In the HITL simulations, all three runway entrance queues were used and multiple runway crossings were set up and used extensively by the local controllers. In the GRP simulation, only one of the three runway entrance queues, the full queue, was used. This was due to limitations of the modeling capabilities of the Surface Operations Simulator and Scheduler (SOSS), which is currently being enhanced. Using the full queue only did not greatly affect the results because its length never extended to the point that it was blocking a taxiway intersection. Furthermore in the GRP simulation, all of the runway crossings ended up being single crossings because there was no logic simulating the actions local controls use to set up multiple crossings. Table 1 compares the 2010 HITL simulation and the GRP simulation.

Table 1. Comparison Between 2010 SARDA HITL and GRP

Simulation Characteristic	Difference/Similarity	2010 SARDA HITL	GRP
airport	similarity	DFW east-side south flow	DFW east-side south flow
traffic scenario	similarity	2008 normal (also had heavy)	2008 normal only
delay-optimal scheduler	similarity	delay-optimal based on DP (used in both RS and SRP)	delay-optimal based on DP (used in GRP)
frequency of scheduler call	difference	12 seconds	5 minutes
clearances	difference	spot	gate
sequencing at runway	difference	delay-optimal	first-available-first-served
runway queues	difference	full, inner, outer	full
crossings	difference	multiple crossings	single crossings
weight classes	difference	small, large, heavy	all large
departure fix constraint	difference	none	80 seconds

III. Fast-time Simulation

This section describes the fast-time simulation tools and setup. It illustrates how the main parts of the system, Surface Operations Simulator and Scheduler (SOSS) and the scheduler, fit together. It also introduces the details of the airport and flight dynamics models.

Figure 1 illustrates how the pieces of the fast-time simulation fit together. SOSS simulates aircraft movement on the airport surface. For each flight in the schedule, it sends the flight's scheduled pushback time (PTime) or wheels-on time, estimated unimpeded time of arrival (ETA) at the runway threshold or gate, and taxi route to the scheduler. Although in the 2010 SARDA concept the scheduler contained two programs: the runway scheduler (RS) and the spot release planner (SRP), in this research the scheduler only contained one program, a gate release planner (GRP). The GRP calculates gate release clearances for each flight based on its departure clearance time. The tactical flight separation (TFS) model simulates actions that pilots and controllers take to keep aircraft safely separated. It is embedded within the SOSS simulation.

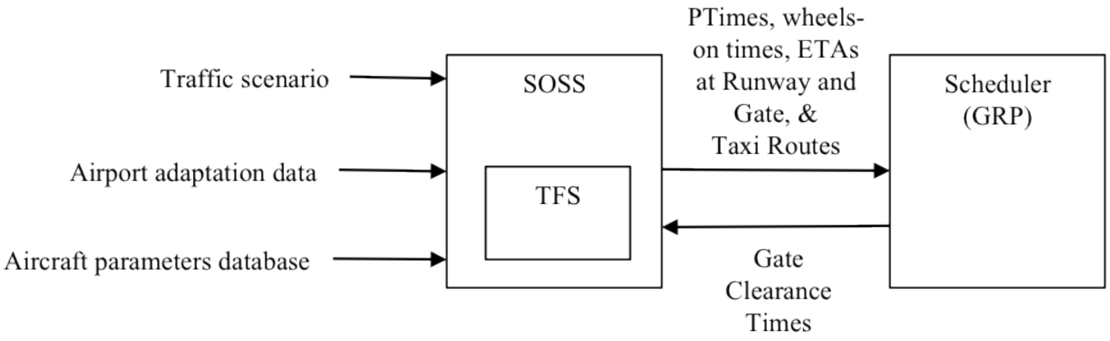
Traffic scenario
Airport adaptation data
Aircraft parameters database
SOSS
TFS
PTimes, wheels-on times, ETAs at Runway and Gate, & Taxi Routes
Scheduler (GRP)
Gate Clearance Times

Figure 1. System Diagram

A. Surface Operations Simulator and Scheduler

SOSS is a fast-time simulation of aircraft movement on the airport surface. It models flight departures and arrivals moving along airport runways, taxiways, and ramps.

SOSS connects with schedulers via NASA's common algorithm interface, which is the same interface used by all of NASA's airport simulation tools. This feature makes SOSS able to easily connect with any of the schedulers that NASA has developed because they also use the common algorithm interface. The common algorithm interface passes for each flight within the current time horizon Ptimes or wheels-on times, routes, and ETAs to the scheduler. The scheduler uses this information to create a new schedule, then passes the schedule back to SOSS in the form of a set of clearance times.

The user controls how SOSS connects to the scheduler. Through SOSS's graphical user interface (GUI), the user can set the frequency of calls to the scheduler and the planning horizon. In this research, the scheduler was called every five minutes and the planning horizon was set to ten minutes. Several simulations were conducted with different call frequency and planning horizons, and these values produced adequate results with good execution speed.

SOSS accepts a traffic scenario file that contains the filed flight schedule. This schedule includes a filed pushback time (Ptime), a taxi route, and a takeoff runway for departure flights and a landing runway, a wheels-on time, a taxi route, and gate for arrival flights. In the simulations, the Ptime is updated by the gate release time calculated by the gate release planner, and the wheels-on times are treated as absolutes that may not be modified. In practice, wheels-on times can be slightly modified by TRACON controllers who are controlling the final approach phase of the flight.

In addition to the traffic scenario file, SOSS also reads in airport adaptation data that describes the node/link network model of the airport and the routes that flights will use to move from the gates to the runways and visa versa. An aircraft database is also read by SOSS. It contains parameters describing important characteristics of each type of aircraft.

For executing a simulation, SOSS contains three models: an airport model, a flight dynamics model, and a tactical flight separation model. These are described in more detail in the following sections.

1. *Airport Model*

SOSS models the ramps, taxiways, and runways at airports with nodes and links. Any airport can be characterized by applying nodes and links to its specific ramp/taxi/runway layout. Nodes represent points on the airport surface, and links represent paths between nodes. Since links are straight lines, curved paths must be approximated with multiple nodes and links. Intersections where multiple paths meet are represented by nodes.

Flight routes are paths through the predefined airport node/link network. They are defined by a set of airport nodes. Flight clearance times at nodes in the routes can be set by the scheduler. When a clearance time is set at a node in a flight's route, SOSS does not allow the flight to exit the node until after the clearance time has elapsed. Although in general a clearance time can be set at any node in a flight's route, in the present research flight clearances for each flight are set only at the gate.

The simulation primarily focused on operations on the east side of Dallas/Fort Worth International Airport (DFW). However, it included a few flights that crossed sides. East- and west-side operations at DFW are almost independent of each other because they are controlled by separate towers and their areas of control are divided down the center of the airport. The two sides connect only by the north and south bridges, which flights use to cross the airport. The airport was in the south flow configuration, which is used about 70% of the time. Jet arrivals approached from the north and landed on runways 17C, 17L, and 18R (18R is on the west side of the airport, not shown in the Fig. 2) and jet departures took off from runways 17R and 18L (18L is also on the west side of the airport) towards the south. Flights departed from and arrived at gates on the west and east sides. The south flow configuration and focus on the east side of DFW is consistent with the SARDA 2010 HITL simulation.

Figure 2 shows the node/link model with the runways labeled and outlined in black and the main east north/south taxiways highlighted in purple. The blowup pictures provide greater detail of the nodes and links. In clockwise direction the blowups are of the runway 17R departure queue, the intersection leading to runway 13L, and Terminal A ramp. Table 2 shows what each of the node colors denotes. Runway 13L is used primarily for turboprops, which where not included in the traffic scenario used in this paper.

Table 2. Node Color Key

Color	Meaning
light blue	departure queue
dark blue	taxiway, gate, or hold
red	runway crossing
green	ramp
yellow	spot
magenta	runway entrance
orange	runway exit

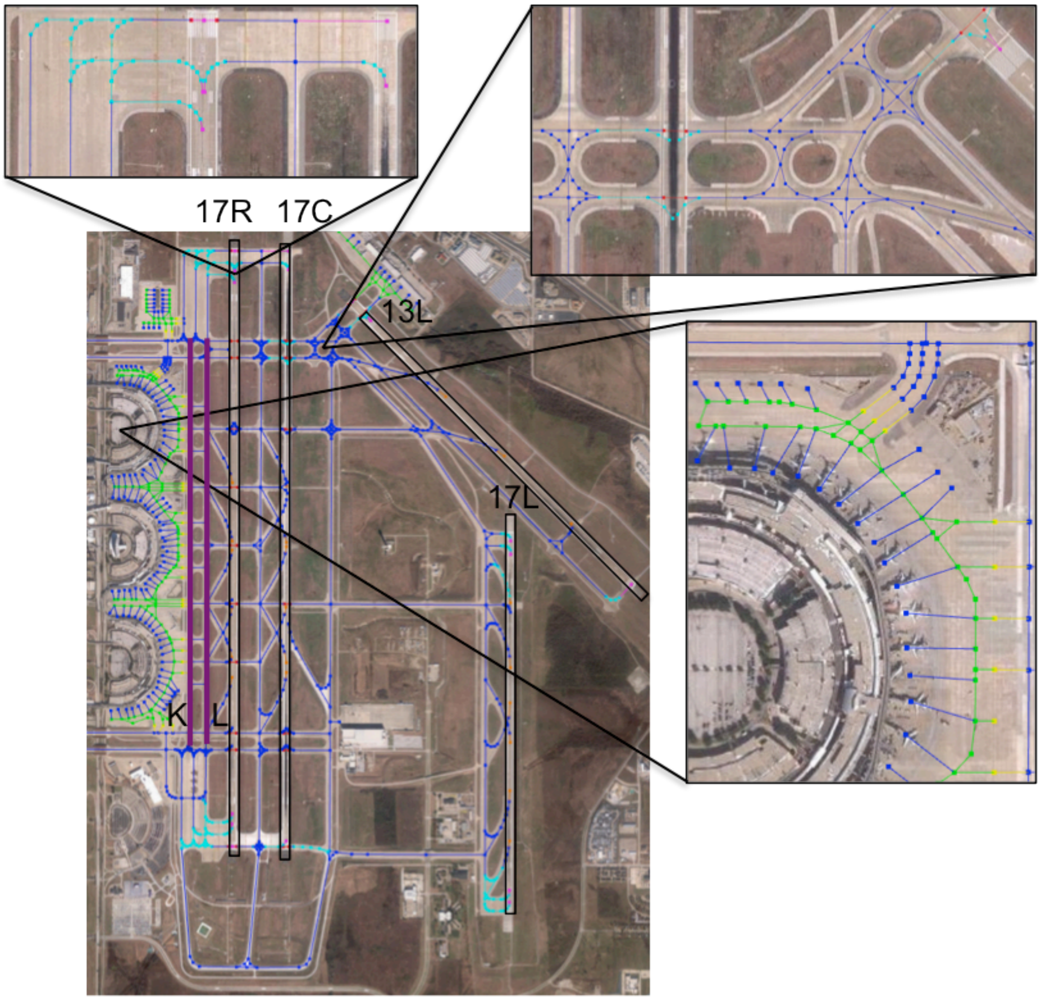

Figure 2. Node/Link Model of East Side of Dallas/Fort Worth International Airport

2. Aircraft Dynamic Model

The aircraft dynamic model determines the movement of the aircraft through the node/link model. A mathematical model defines the aircraft dynamics and a control model defines how each aircraft attempts to achieve its target speed.

The aircraft dynamic model uses one-degree-of-freedom kinematic equations. The equations are

$$\frac{ds}{dt} = v,\ \frac{dv}{dt} = a,\ \text{where } a \doteq \begin{cases} a_{\max} \\ 0 \\ a_{\min} \end{cases} \tag{1}$$

where s is path length, v is aircraft speed, and a is aircraft acceleration. Acceleration is defined as one of three values depending on the current and target speeds of the aircraft. If the current speed is equal to the target speed, a=0. If the current speed is less than the target speed, a=$a_{\max}$, and, if the current speed is greater than the target speed, a=$a_{\min}$, which is a negative value. An aircraft has three different target speeds: one for movement in the ramp area (v_{ramp}), one for movement in the taxiways (v_{taxi}), and one for movement in the runway queues (v_{queue}).

The accelerations and the target speeds are defined in the aircraft parameters database. This database contains these and other parameters for 458 different types of aircraft. Table 3 gives the parameters for a Boeing 737 jet. These values are typical for the majority of the jets in the simulations.

Table 3. Aircraft Dynamic Model Parameters for a Boeing 737 Jet

Parameter	Value
a_{max}	1.85 ft/sec^2
a_{min}	-1.85 ft/sec^2
v_{queue}	10 kts
v_{taxi}	15 kts
v_{ramp}	10 kts

The target speed of the aircraft is determined by the aircraft speed control model. SOSS has two speed control models: open-loop and closed-loop. Open loop speed control simulates flight procedures used in the current system, while closed loop speed control simulates flight procedures in a future system.

In open-loop control, the aircraft's target speed is determined from the aircraft parameters database. This target speed represents the preferred taxi speed that a pilot would use. As described above, this speed is constant for aircraft of the same type and in the same area: departure queue, taxiway, or ramp. In the real system, different pilots taxi at different speeds. The target speed used in SOSS represents an average of the distribution of speeds used in real life. In the present simulations, target speeds were treated as deterministic.

Closed-loop control overrides the target speed from the database and adjusts it within bounds given by the database such that the flights rendezvous with their target nodes at just the scheduled times. For this to occur in practice, aircraft would require new flight deck equipment.

The simulations in the current research were meant to model existing operations. Therefore, the open loop control option was used.

3. Tactical Flight Separation Model

The tactical flight separation model simulates tactical actions that pilots and controllers take to maintain safe separation between aircraft. Separation is handled differently for flights using the runway than for flights taxiing through the ramp, taxi, and queuing areas.

Tactical separations at the runway are determined by navigational safety and wake-vortex spacing constraints. In practice, controllers enforce a separation using a distance-based rule. SOSS enforces a separation by holding a flight at the entrance of the runway until a specified amount of time from the previous operation has elapsed. The time is calculated to achieve the correct distance between operations. Table 4 shows the times required between consecutive departures. The weight classes of the aircraft are denoted by small (S), large (L), heavy (H), and B757 (B75). The B757 has its own weight class because its weight is in the large class, but it creates more turbulence than other large aircraft. Rows specify the weight class of the leader, and columns specify the weight class of the follower. The separation time between a departure and a crossing, and vice versa, was 40 seconds.

Table 4. Separation in seconds between consecutive departures

	S	L	H	B75
S	45	68	82	82
L	45	45	68	68
H	45	45	82	68
B75	45	45	82	68

In SOSS, separation at the departure fix was also constrained. SOSS actually enforces the separation constraint at the runway take-off point because it currently does not model movement of aircraft in the air. In these simulations, consecutive flights headed for the same departure fix were spaced by 80 seconds. A spacing of 80 seconds at the departure fix translates into a five miles-in-trail restriction into the enroute airspace. This was not consistent with the SARDA 2010 HITL simulation. That simulation did not have constraints at the fixes.

The method used by SOSS for maintaining separation in the ramp, taxi and queuing areas consists of a conflict-detection algorithm and a conflict-resolution algorithm. In contrast to the tactical separation model, the gate release planner did not attempt to give clearance times at the gate that would prevent conflicts. It was assumed that conflicts would be taken care of by the tactical separation algorithm. The conflict-detection algorithm uses the aircraft route to project the locations of all aircraft at future times, which are configurable by the user. In this research, 35.7, 75, 112.5, and 150 seconds in the future were selected because they allowed the detection algorithm to find most conflicts and did not noticeably slow SOSS execution speed. The projected locations of each aircraft are checked against the projected locations of the other aircraft to determine if there are any pairs of aircraft that will get too close to each other. If a pair is identified, the situation is called a conflict and actions are taken to prevent the two aircraft from getting too close.

Conflicts are classified into one of several categories: in-trail, head-on, and merging. An in-trail conflict occurs when a trailing aircraft overtakes its lead traveling in the same direction on the same link, and a head-on conflict occurs when two aircraft traveling in different directions on the same link violate separation. A merging conflict occurs when two aircraft cross paths on an intersection at the same time.

The resolution algorithm generates actions that prevent the conflict from turning into an actual collision. The resolution algorithm handles in-trail, merging, and head-on conflicts differently. For in-trail conflicts, the algorithm slows down the trailing aircraft. For merging conflicts, the algorithm determines which aircraft will arrive at the intersection first. Then, it slows the other aircraft so that the first can safely pass through the intersection. Finally, for head-on conflicts, the algorithm determines the set of links that make up the common path between the aircraft. Then, it identifies which aircraft will arrive first at its nearest link in the set. The aircraft that reaches its nearest link last is slowed or stopped outside the common path until the other aircraft safely passes.

B. Scheduler

The GRP produced gate-release times for flights using a two-step process. First using a delay-optimal scheduler, it built a runway take-off schedule. The delay-optimal scheduler was the same as that used in the core of the RS and SRP in the 2010 SARDA HITL, and it was based on a dynamic programming approach. The details of the dynamic programming approach are published in Refs. 11 and 12. The schedule was constrained so that it satisfied wake vortex, navigation, and miles-in-trail separation requirements.

In the second step, the gate-release time was calculated by subtracting the unimpeded taxi-time from the flight departure time. This clearance time did not account for traffic that the flight would encounter on its way to the runway. The tactical flight separation model was used to resolve conflicts that occured.

IV. Results

Two fast-time simulations of normal traffic at Dallas/Fort Worth International Airport were conducted to show the benefits of the gate release planner (GRP). These simulations illustrate how SOSS can be used to analyze the benefits of a concept for managing flights on the airport surface. The simulations are not an analysis of SARDA itself due to the differences listed in Section II. The GRP was used to control surface traffic in one simulation, and a first-come-first-served (FCFS) scheduler was used to control traffic in the other simulation. Performance of the GRP scheduler was compared with that of the FCFS scheduler. The FCFS scheduler was used for comparison because it was simple to model. It represents the case where operations are conducted in the same sequence that they are scheduled. Although it is similar to real operations, it is not the same because typically a controller will do position-constrained re-sequencing to increase throughput.

The departure spacing restrictions listed in Table 4 were enforced by SOSS. Similarly, 40 seconds were maintained between departures and crossings. Finally, an 80 second constraint was applied to consecutive departures using the same departure fix.

In the GRP simulation, minimum-delay sequences of departures on runway 17R were generated by the optimal scheduler. Arrivals from 17C and 17L that needed to cross 17R were entered into the sequence on an availability basis. When there was a crossing that was ready to use 17R before the leading departure, it was cleared to cross. The schedules were updated every five minutes, starting at the initial simulation time of zero. Generally a higher update rate would be desired to capture as soon as possible any events that were not predicted. Five minutes was used here because the only unpredicted events were conflicts, and there were not enough conflicts to adversely affect the performance of the scheduler. The time horizon of the schedules was ten minutes. Time horizons that are too large can cause the scheduler to execute slowly because there are too many flights in the schedule. A ten-minute horizon caused the scheduler to execute quickly, less than a second, while not degrading the benefits of the simulation. Operations on runways other than 17R were sequenced using first-come-first-served.

As discussed in the Section II, flights in the GRP simulation were controlled by strategically releasing them from their gates. No other strategic clearances were given to the flights as they moved through the ramp, taxi, and queue areas. However, as described in the tactical separation section of the paper, the flights did use tactical maneuvers to avoid separations violations. This caused the flights to actually reach the runway entrance at uncertain times that could differ from the runway-scheduled times. No other uncertainties were modeled in the simulation.

A. Traffic Scenario

The traffic scenario was created by recording live surface traffic at Dallas/Fort Worth International for a peak hour in 2008. All flights with routes that did not intersect with runway 17R or the main inboard north/south taxiways K and L (shown in purple on Fig. 2) were filtered out. This produced full traffic levels in these areas, while minimizing the number of flights that were simulated.

The traffic scenario consisted of 80 flights scheduled in a 42 minute period. Forty flights were arrivals and forty flights were departures. All of the flights were in the large weight class.

Figure 3 illustrates the 17R operation departure rates. There are three curves. The demand curve represents the operation rate that would occur if the schedule were exactly followed and there were no separation constraints at the runway or along the taxiways. The FCFS curve illustrates the actual operations rates in the FCFS simulation, and the GRP curve illustrates the actual operations in the GRP simulation. The FCFS is drawn mostly on top of the GRP curve, except for a small portion from 700 to 1300 seconds. The operations rates were calculated by counting the number of departures and crossings occurring within a 5 minute sliding bin. Each point in the curve is created by shifting the bin one minute.

The runway capacity is the operation rate where the runway cannot accept more operations without violating a separation constraint. It is shown on Fig. 3 where the FCFS and GRP curves flatten out. The capacity for 17R is about 7.5 operations per 5 minutes, depending on the mix[†] of operations. As shown in Fig. 3, demand exceeds capacity from approximately 500 to 900 seconds and from 1250 to 1750 seconds in the traffic scenarios.

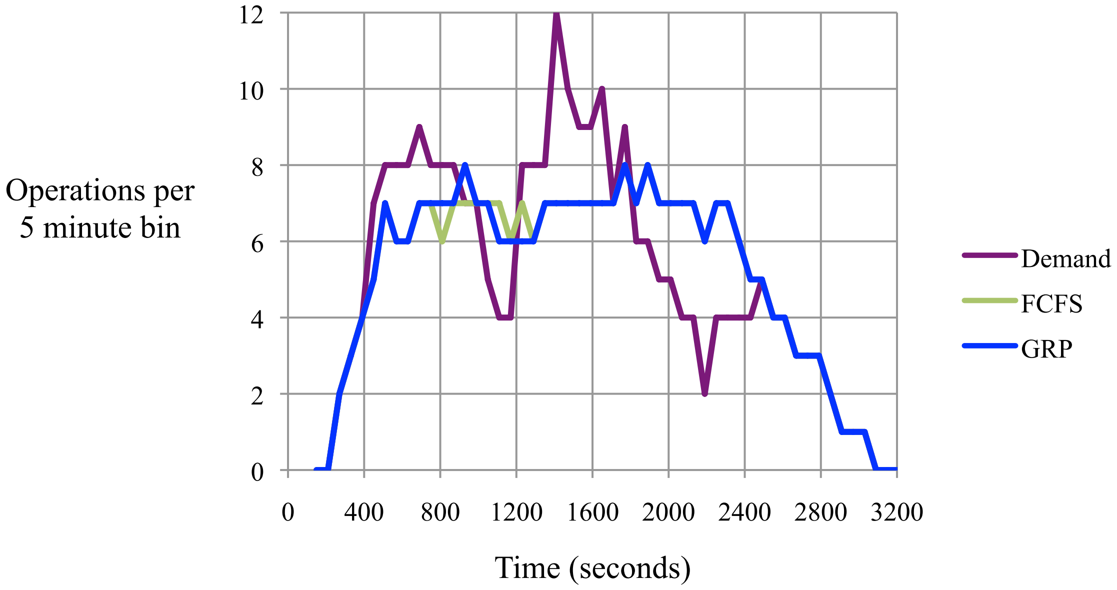

Figure 3. Runway 17R Operation Rates in Number of Departures and Crossings per Five-Minute Bin

[†] Arrival and departure operations take different amounts of time, 40 seconds and 45 seconds respectively, and consecutive departures to the same fix take 80 seconds. The mix of operations will affect the capacity of the runway.

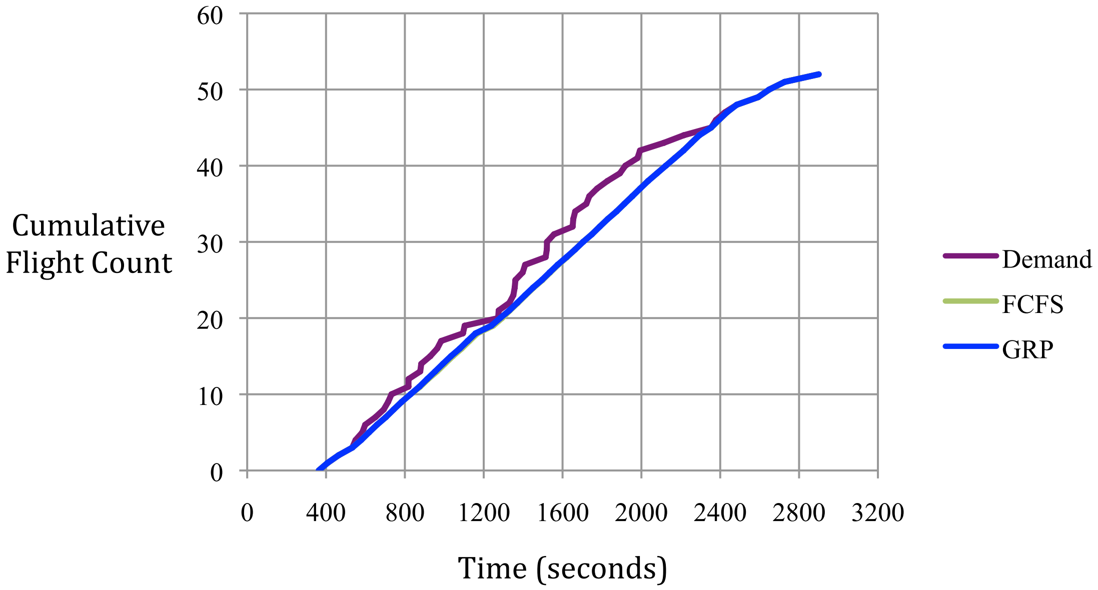

Figure 4. Runway 17R Cumulative Operations vs. Simulation Time

In Fig. 3, the FCFS and GRP curves are higher than the demand curves from approximately 1000 to 1250 seconds and 1750 to 2500 seconds. These intervals represent times when 17R departures have been delayed, and, even though demand has dropped, the actual operation rate is still at capacity so that the waiting flights can depart.

Figure 4 shows the cumulative operations verses simulation time. Cumulative operations is like operations per bin except that a bin size does not need to be selected to create the plot. Again, the GRP curve is on top of the FCFS curve. At time 1300 the GRP and FCFS curves touch the demand curve, indicating that the backlog of flights waiting to use 17R has been served. After 1300 seconds the demand increases beyond the capacity of 17R and a backlog accumulates again. That backlog is finished be served at 2400 seconds. The maximum capacity of 17R is shown on this plot by the maximum slop of the line.

There is very little difference between the GRP and FCFS curves. This indicates that in the GRP simulation the optimal scheduler was not able to find a sequence of runway operations that was able to increase the runway throughput above the level achieved by first-come-first-served. This is the result of only one simulation, more simulations would be needed to establish this result in general. The optimal scheduler may have had more opportunity to find better sequences if crossings had been included in the schedule and the inner and outer queues had been used. Also, all of the flights in the traffic scenario were in the large weight class. If there had been a more diverse set of weight classes, the scheduler would have had more opportunities to find better sequences.

B. Metrics

In this section, three metrics are explored: delay, number of stops, and taxi times. Delays were calculated by subtracting nominal unimpeded times from actual times. For example, to calculate total departure delay for a flight, the nominal time at which the flight was initially planning to takeoff is subtracted from the actual time the flight took off. These times can be taken at any node along the route, not just the take-off node. The number of stops is calculated by counting the number of times the speed of the aircraft drops below a specified threshold value. In this research, 0.1 knot was used as the threshold value because it is 100 times smaller than the target speed in the queue area, which was 10 knots. Taxi times were calculated by subtracting the release time at the gate from the take-off time at the runway.

Table 5 gives averages, standard deviations, minimums and maximums of each metric for departures in the two simulations. Table 6 does the same for arrivals. Average delays are very similar for departures in the two simulations. This means the GRP was not able to reduce the total departure delay for flights. However, the average taxi time for departures in the GRP simulation was 45 seconds less than that for the FCFS simulation. This is because in the GRP simulation departures were held longer at their gates. Since engines are not turned on while at the gates, this lower taxi time directly translates into less fuel burned, emissions, and noise. Similarly, departures in the GRP simulation had fewer average stops. This is because by holding departures at the gates, the GRP reduces the queue of departures waiting at the runway entrance. Departures that wait in line for less time have to stop and go less often.

According to Table 6, the GRP arrivals had a little less average delay. The slight difference is because holding flights at the gates produces less traffic for the arrivals to navigate through to get to their gates. Similarly, the GRP arrivals had fewer stops than the FCFS arrivals. In addition, the standard deviations are large. Thus, the less delay for GRP arrivals is not consistent.

Table 5. Departure Delays, Stops, and Taxi Times

	Delay (sec)	Number of Stops	Taxi Time (sec)
	FCFS / GRP	FCFS / GRP	FCFS / GRP
Average	123 / 122	20 / 12	537 / 492
Standard Deviation	105 / 111	22 / 15	140 / 131
Minimum	0 / 0	1 / 1	256 / 256
Maximum	343 / 428	67 / 50	798 / 708

Table 6. Arrival Delays, Stops, and Taxi Times

	Delay (sec)	Number of Stops	Taxi Time (sec)
	FCFS / GRP	FCFS / GRP	FCFS / GRP
Average	21 / 18	3 / 1	444 / 440
Standard Deviation	24 / 23	6 / 0.5	217 / 216
Minimum	0 / 0	1 / 1	154 / 154
Maximum	76 / 76	36 / 2	870 / 870

Note that in Tables 5 and 6, flights are not necessarily consistent across the maximum and minimum rows. For example in Table 6, the flight with 36 stops was not the same as the flight with a taxi time of 870 seconds.

Since SOSS categorizes the airport nodes and links into gate, ramp, spot, taxi, queue, and departure, the metrics can be broken down into these categories. Table 7 shows the departure metrics broken out by airport area. Taxi time is not applicable for the gate because it is defined as time from gate release to take-off.

In the FCFS simulation, there is little delay in the gate, ramp, and spot sections of the airport. In the GRP simulation however, there is more delay at the gate because flights were strategically held at the gate by the scheduler. These delays at the gate lead to less delay in the taxi, queue, and departure areas. This shows that the queue of flights waiting to use 17R was smaller in the GRP simulation. A similar trend is seen in the number of stops data. Finally, the FCFS taxi times were higher in the queue area than the GRP taxi times. This again was caused in part by the GRP queue length being shorter.

Although the length of the departure queue was not calculated in the simulations, it was visible in the simulation display. By visual inspection, the peak queue length in the GRP simulation was about five flights and in the FCFS simulation was about seven flights.

Table 7. Departure Metrics broken out by Airport area

	Gate	Ramp	Spot	Taxi	Queue	Runway
	FCFS / GRP	FCFS / GRP	FCFS / GRP	FCFS / GRP	FCFS / GRP	FCFS / GRP
Delay	0 / 43	4 / 4	0 / 1	25 / 4	66 / 44	29 / 27
# of Stops	0 / 0	0 / 0	0 / 0	1 / 0	18 / 11	1 / 1
Taxi Time	N/A	75 / 76	0 / 0	299 / 279	134 / 112	29 / 27

In addition to efficiency, equity is also important. Greater throughput should not be achieved by excessively delaying a few flights. The equity of each simulation is illustrated by a histogram of the flight delays. If the tail of the histogram is thin and long, a small number of flights were excessively delayed. Figure 5 shows a delay histogram for the simulations. The GRP simulation has a longer tail. According to Table 5, the worst delay in the GRP simulation was 428 seconds, and the worst delay in the FCFS simulation was 343 seconds. The difference between the two is about one and half minutes.

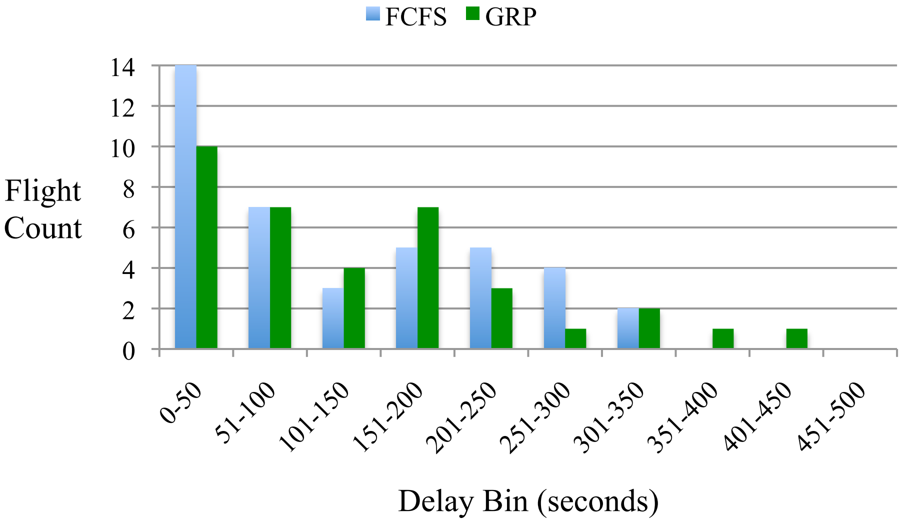

Figure 5. Histogram of Departure Delays for FCFS and GRP Simulations

V. Future Work

The GRP simulation did not measurably increase the throughput rate of runway 17R or decrease average departure delay despite other research[6-12] that shows in standalone analysis that optimal departure scheduling can decrease delay. Future research could investigate other traffic scenarios, including runway crossings in the optimal scheduling process, and better prediction and control methods to see if optimal scheduling can increase the runway throughput.

The GRP simulation was different from the SARDA 2010 HITL. To better confirm the results of SARDA 2010, the fast-time simulations could be made more like SARDA by including runway crossings in the delay-optimal scheduler, routing departures through the full, inner, and outer queues which feed runway 17R, and enforcing an optimal sequence at runway 17R. In addition, the GRP simulation used flight clearances at the gates. The 2010 SARDA simulation used flight clearances at the spots. A study could be conducted of how clearing flights at the spot or the gate would change the benefits.

Aside from conflict detection and resolution, this study did not include uncertainty. Future research could include uncertainty in gate pushback times and uncertainty in human response times.

This study focused on a single runway at Dallas/Fort Worth International Airport. A future study could apply the optimal scheduling to more than one departure runway, including runways with mixed operations or constraints with neighboring runways.

In this study only a single traffic scenario was simulated. A future study could investigate many traffic scenarios.

VI. Conclusion

A fast-time simulation of airport surfaces, the Surface Operations Simulator and Scheduler (SOSS), was able to analyze the benefits of a delay-optimal scheduler controlling surface traffic on the east side of Dallas/Fort Worth International airport. The delay-optimal scheduler used in the fast-time simulation was adapted from the 2010 Spot and Runway Departure Advisor (SARDA) human-in-the-loop (HITL) simulation. Although the fast-time simulation was meant to measure the benefits of the SARDA concept, differences between the fast-time and HITL simulations made it difficult to exactly compare the results. The differences were due to time and SOSS modeling limitations that are being resolved. It is desirable to fix these issues in SOSS so that it can be used to analyze SARDA against more traffic scenarios and changes than would be possible with HITL simulation alone.

The delay optimal fast-time simulation had 8% less taxi time and 40% fewer stops relative to a simulation of a first-come-first-served scheduler managing traffic. The general trend of reduced taxi times and number of stops was also observed in the heavy traffic scenario of the 2010 SARDA HITL simulation. This is despite the differences between the HITL and fast-time simulations. In general, the reduced taxi times and fewer stops are due to the delay-optimal scheduler holding flights at the gate or spot longer. This reduces traffic in the ramp and taxiways and shortens the length of the departure queue.

The average delay of the delay-optimal simulation was not less than that of the first-come-first-served simulation. This is in contrast to stand-alone analysis of delay-optimal schedulers,[6-12] which showed that they reduce delays and increase throughput relative to simple algorithmic models of human controllers.

Equity of flight delays was also studied. It is not equitable for a few flights to be excessively delayed so that the other flights can enjoy less or no delays. The largest flight delay in the delay-optimal simulation was one and a half minute longer than the largest flight delay in the first-come-first-served simulation.

The results of this paper need to be strengthened by more simulations with a variety of traffic scenarios. In addition, changes to the fast-time simulation such as including runway crossings in the delay-optimal departure schedule could reduce the average flight delay.

Acknowledgments

The author would like to acknowledge Zhifan Zhu for his dedicated assistance with debugging and executing SOSS and Justin Montoya for explaining how the runway scheduler works.

References

[1] Jung, Y., Hoang, T., Montoya, J., Gupta, G., Malik, W., Tobias, L., and Wang, H., "Performance Evaluation of a Surface Traffic Management Tool for Dallas/Fort Worth International Airport," *Ninth USA/Europe Air Traffic Management Research and Development Seminar*, 92, Berlin, Germany, 2011.

[2] Zelinski, S., and Romer, T., "An Airspace Concept Evaluation System Characterization of National Airspace System Delay," 4th AIAA Aviation, Technology, Integration, and Operations Conference, AIAA 2004-6200, Chicago, IL, 2004.

[3] Simaiakis, I., Khadilkar, H., Balakrishnan, H., Reynolds, T., Hansman, J., Reilly, B., and Urlass, S., "Demonstration of Reduced Airport Congestion Through Pushback Rate Control," *Ninth USA/Europe Air Traffic Management Research and Development Seminar*, , Berlin, Germany, 2011.

[4] Khadilkar, H., and Balakrishnan, H., "Analysis of Airport Performance using Surface Surveillance Data: A Case Study of BOS," *11th AIAA Aviation, Technology, Integration, and Operations Conference*, AIAA 2011-6986, Virginia Beach, VA, 2011.

[5] Brinton, C., Atkins, S., Cook, L., Lent, S., and Prevost, T., "Ration by Schedule for Airport Arrival and Departure Planning and Scheduling," *2010 Integrated Communications Navigation and Surveillance Conference*, Herndon, VA, 2010.

[6] Gupta, G., Malik, W., and Jung, Y., "A Mixed Linear Program for the Airport Departure Scheduling Problem, "AIAA Aviation, Technology, Integration, and Operations Conference, AIAA 2010-7692, Hilton Head, SC, 2009.

[7] Gupta, G., Malik, W., and Jung Y., "Incorporating Active Runway Crossings in Airport Departure Scheduling," *AIAA Guidance, Navigation, and Control Conference*, AIAA 2010-7695, Toronto, Canada, 2010.

[8] Malik, W., Gupta, G., and Jung, Y., "Managing Departure Aircraft Release for Efficient Airport Surface Operations," AIAA Guidance, Navigation, and Control Conference, AIAA 2010-7696, Toronto, Canada, 2010.

[9] Montoya, J., Zachary, W., Rathinam, S., and Malik, W., "A Mixed Integer Linear Program for Solving a Multiple Route Taxi Scheduling Problem," AIAA Guidance, Navigation, and Control Conference, AIAA 2010-7692, Toronto, Canada, 2010.

[10] Gupta, G., Malik, W., and Jung, Y., "Effect of Uncertainty on Deterministic Runway Scheduling," AIAA Aviation Technology, Integration, and Operations Conference, AIAA 2011-6924, Virginia Beach, VA, 2011.

[11] Rathinam, S., Wood, Z., Sridhar, B., and Jung, Y., "A Generalized Dynamic Programming Approach for a Departure Scheduling Program," *AIAA Guidance, Navigation, and Control Conference*, AIAA 2009-6250 Chicago, IL, 2009.

[12] Montoya, J., "Runway Scheduling Using Generalized Dynamic Programming," *AIAA Guidance, Navigation, and Control Conference*, AIAA 2011-6380, Portland, OR, 2011.

[13] Jung, Y., Hoang, T., Montoya, J., Gupta, G., Malik, W., and Tobias, L., "A Concept and Implementation of Optimized Operations of Airport Surface Traffic," *10th AIAA Aviation, Technology, Integration, and Operations Conference*, AIAA 2010-9213 Fort Worth, TX, 2010.

[14] Hoang, T., Jung, Y., Holbrook, J., and Malik, W., "Tower Controllers' Assessment of the Spot and Runway Departure Advisor (SARDA) Concept," *Ninth USA/Europe Air Traffic Management Research and Development Seminar*, 34, Berlin, Germany, 2011.

[15] Wood, Z., Kistler, M., Rathinam, S., and Jung, Y., "A Simulator for Modeling Aircraft Surface Operations at Airports," AIAA Modeling and Simulation Technologies Conference, AIAA 2009-5912, Chicago IL, 2009.